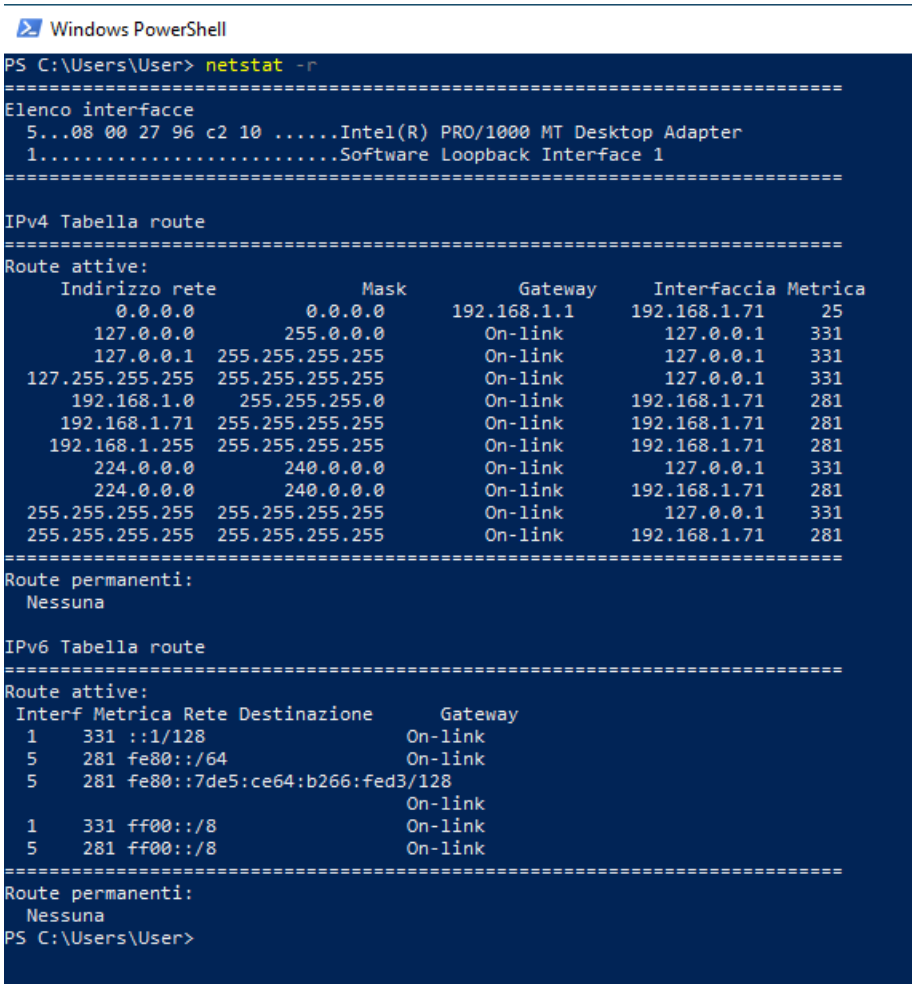
Report Tecnico:
Utilizzo di Windows PowerShell

Autore: Josh Van Edward D Abanico
Data: 20 febbraio 2026



Obiettivo: Esplorazione Windows PowerShell

Indice

1 Introduzione e Scenario **2**
1.1 Obiettivi 2
1.2 Ambiente di Test 2

2 Accesso e Comandi Base **2**
2.1 Avvio dell'ambiente e comandi standard 2
2.2 Esplorazione dei Cmdlet 4

3 Analisi di Rete tramite Netstat **4**
3.1 Tabella di Routing 4
3.2 Correlazione Processi e Connessioni TCP 5

4 Automazione delle Operazioni **7**

5 Domanda di Riflessione **7**

6 Conclusioni **8**

1 Introduzione e Scenario

1.1 Obiettivi

L'obiettivo di questo laboratorio è esplorare alcune delle funzioni di PowerShell. PowerShell è un potente strumento di automazione. È sia una console di comando che un linguaggio di scripting. In questo scenario pratico, la console viene utilizzata per eseguire alcuni dei comandi disponibili sia nel prompt dei comandi che in PowerShell.

1.2 Ambiente di Test

L'infrastruttura di laboratorio è composta da:

- **Sistema Operativo:** 1 PC Windows.
- **Prerequisiti:** PowerShell installato e accesso a internet.

2 Accesso e Comandi Base

2.1 Avvio dell'ambiente e comandi standard

La fase di analisi è iniziata con l'accesso alla console PowerShell tramite il menu Start del sistema. Sono stati testati i comandi di base per l'esplorazione del file system e della rete. Inizialmente, è stato eseguito il comando `dir` al prompt in entrambe le finestre per elencare le directory.

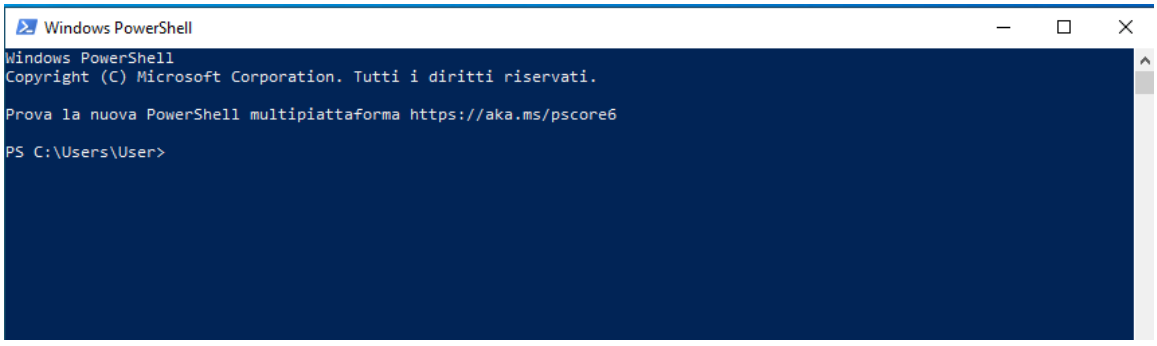


Figura 1: Avvio della console Windows PowerShell.

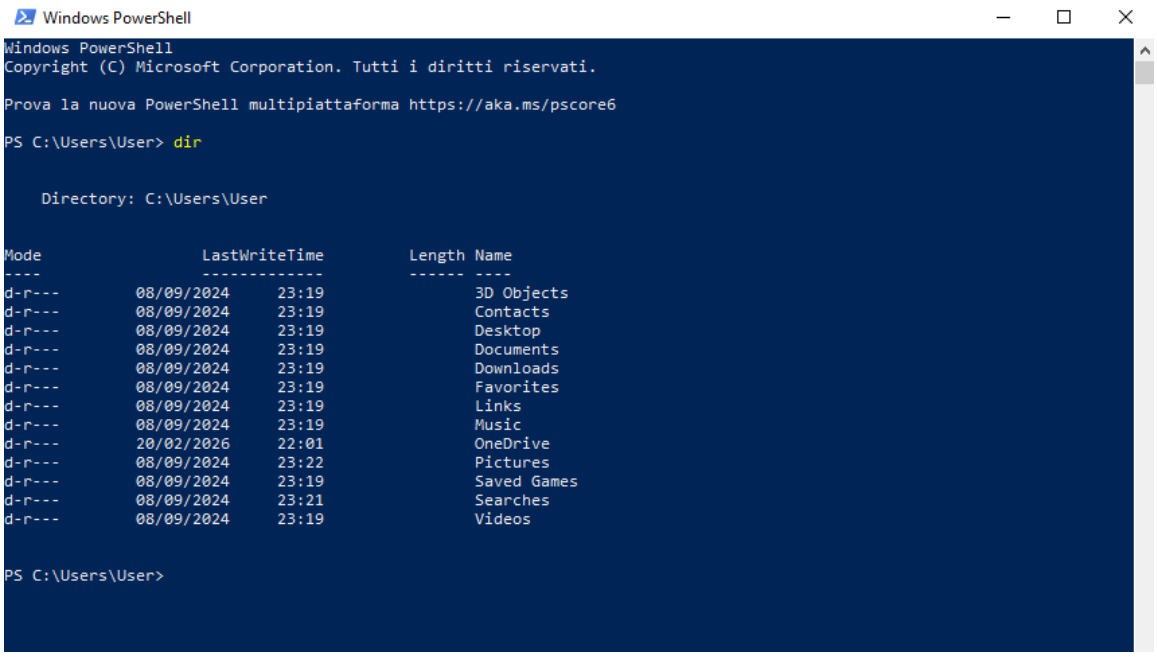


Figura 2: Esecuzione del comando dir per elencare le directory.

Successivamente, la connettività di rete e la configurazione delle interfacce sono state verificate testando un altro comando usato nel prompt dei comandi, come `ping`, `cd` e `ipconfig`.

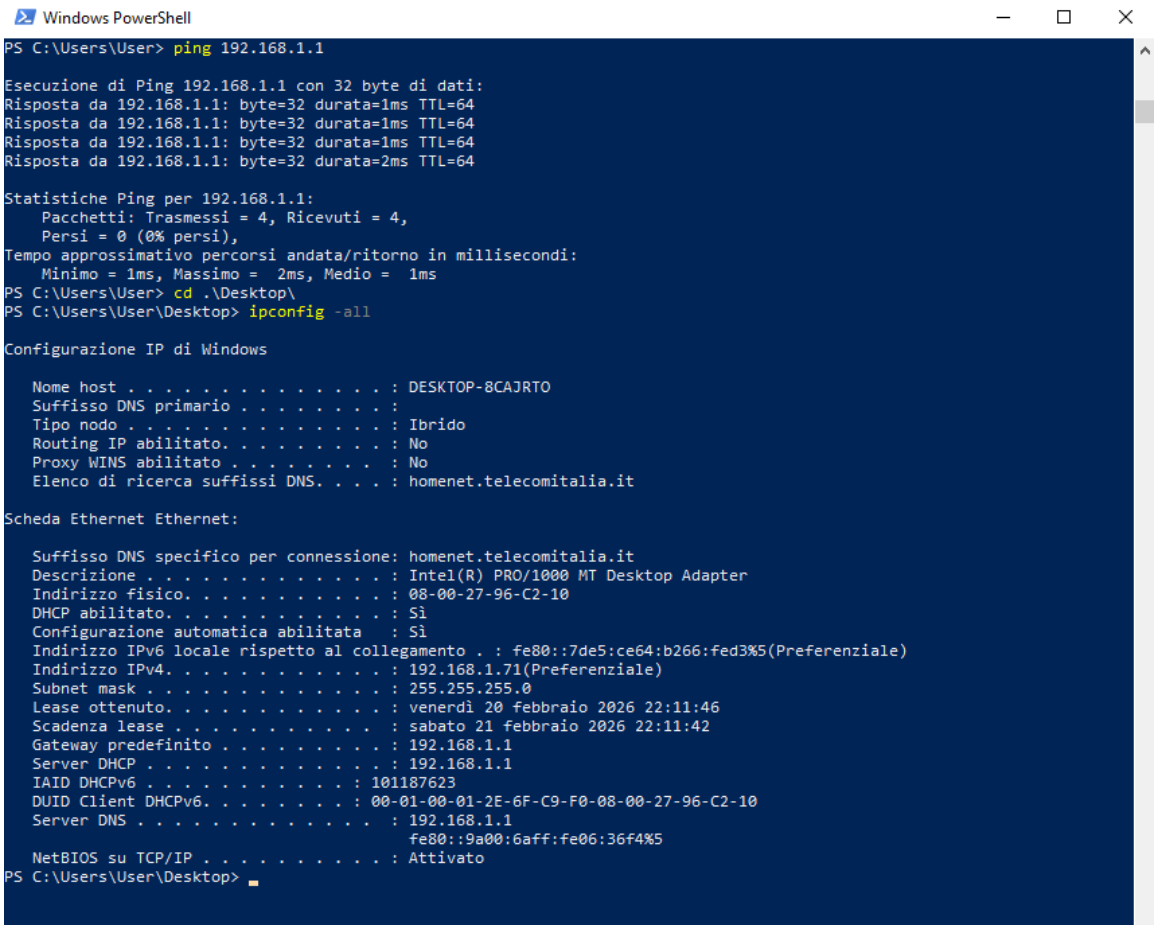


Figura 3: Risultati dei comandi ping, navigazione directory e ipconfig -all.

2.2 Esplorazione dei Cmdlet

I comandi PowerShell, chiamati cmdlet, sono costruiti nella forma di una stringa verbo-nome. Per identificare il comando PowerShell per elencare le sottodirectory e i file in una directory, è stato inserito `Get-Alias dir` al prompt di PowerShell, il quale ha confermato che l'alias `dir` punta a `Get-ChildItem`.

3 Analisi di Rete tramite Netstat

L'esplorazione dello stato della rete è proseguita inserendo `netstat -h` per vedere le opzioni disponibili.

3.1 Tabella di Routing

Per visualizzare la tabella di routing con le rotte attive, è stato inserito `netstat -r` al prompt.

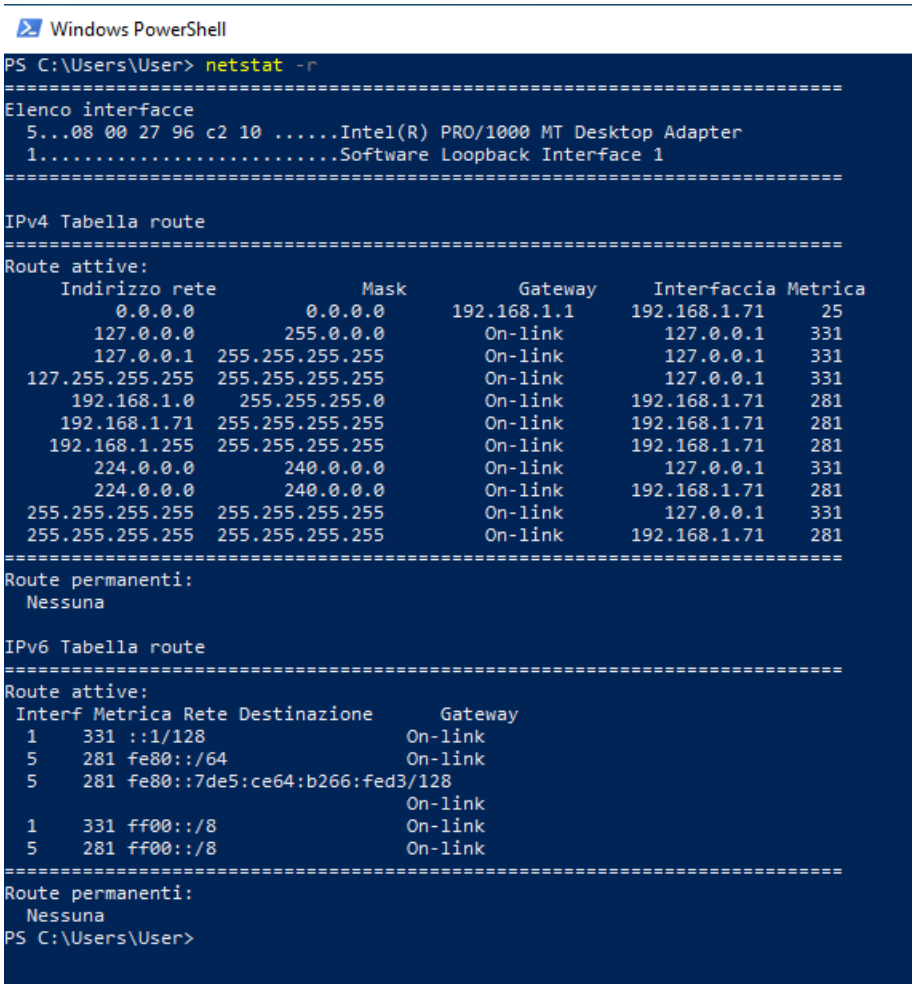


Figura 4: Output del comando netstat -r indicante le rotte attive e le metriche.

3.2 Correlazione Processi e Connessioni TCP

Al fine di eseguire un'analisi più approfondita, è stata aperta ed eseguita una seconda PowerShell con privilegi elevati. Inserendo `netstat -abno` al prompt, il comando `netstat` ha visualizzato i processi associati alle connessioni TCP attive. Parallelamente, è stato aperto Gestione Attività (Task Manager) alla scheda Dettagli (Details) facendo clic sull'intestazione PID in modo che i PID fossero in ordine.

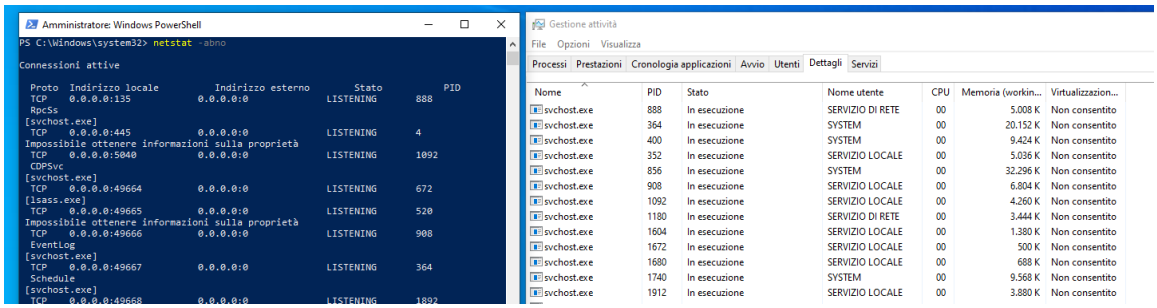


Figura 5: Esecuzione di netstat -abno e correlazione con il PID in Gestione Attività.

È stato poi individuato in Gestione Attività il PID selezionato dai risultati (in questo caso PID 888 associato a `svchost.exe` in `LISTENING`) ed è stato fatto clic con il pulsante destro sul PID selezionato in Gestione Attività per aprire la finestra di dialogo Proprietà (Properties) per maggiori informazioni.

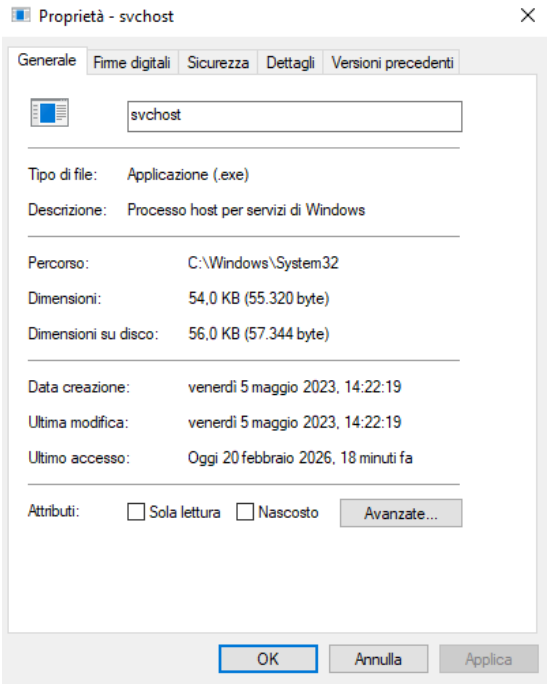


Figura 6: Generale: Percorso e dimensioni.

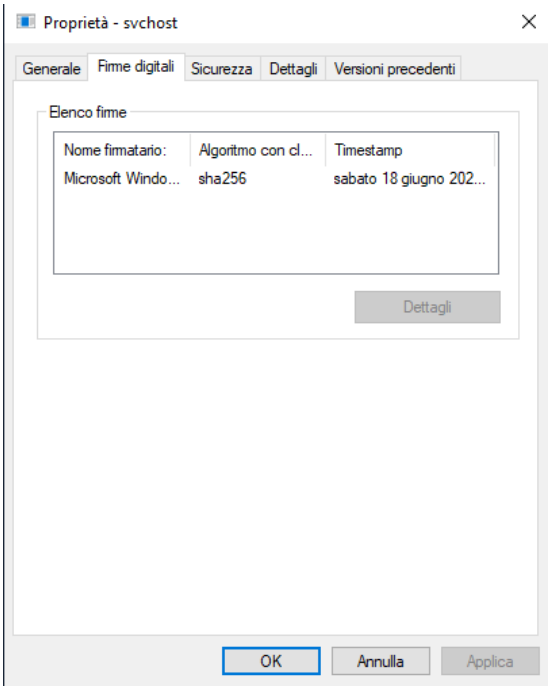


Figura 7: Firme digitali: Verifica firma.

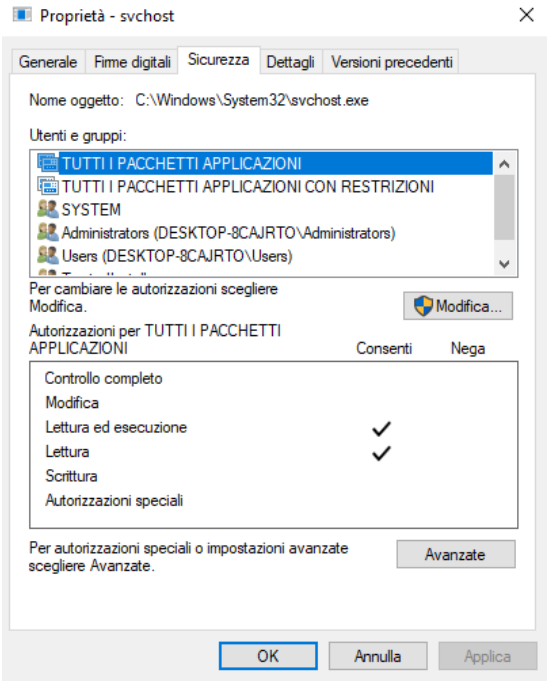


Figura 8: Sicurezza: Autorizzazioni utenti.

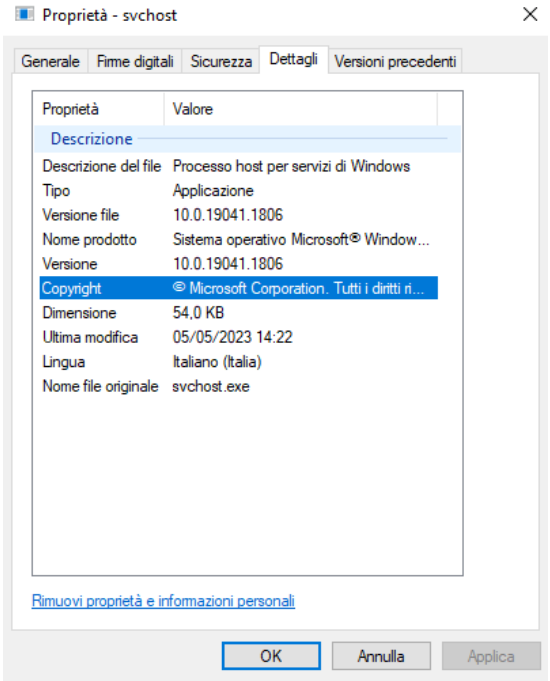


Figura 9: Dettagli: Versione e metadati.

4 Automazione delle Operazioni

I comandi PowerShell possono semplificare azioni che richiederebbero più passaggi per essere eseguite usando gli strumenti grafici del desktop di Windows. In questo passaggio è stata testata la gestione automatizzata del Cestino di sistema.

In una console PowerShell eseguita come amministratore, è stato inserito il comando `Clear-RecycleBin` al prompt. Il sistema ha restituito un avviso di conferma per l'eliminazione permanente di tutto il contenuto del Cestino. Inserendo `s` (Sì), l'operazione è stata completata. Il cambiamento di stato è riscontrabile anche a livello di interfaccia grafica, dove l'icona del Cestino è passata dallo stato pieno allo stato vuoto.

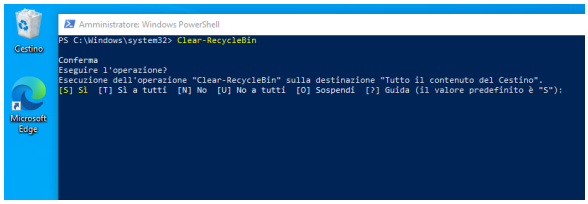


Figura 10: Avviso di conferma per lo svuotamento.

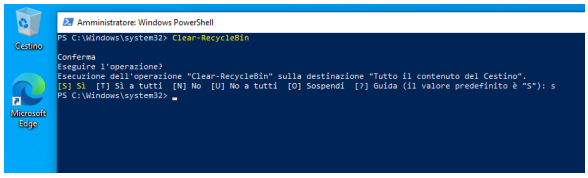


Figura 11: Esecuzione completata e cestino vuoto.

5 Domanda di Riflessione

Come richiesto dall'esercizio, è stata effettuata una breve ricerca per individuare comandi PowerShell (cmdlet) in grado di semplificare e automatizzare le attività quotidiane di un analista di sicurezza. Di seguito si registrano le scoperte principali:

- `Get-WinEvent`: Un comando essenziale per l'analisi dei log. Permette di interrogare, filtrare e analizzare rapidamente gli eventi di sicurezza di Windows (ad esempio, individuare i tentativi di login falliti filtrando l'Event ID 4625) in modo molto più veloce rispetto all'interfaccia grafica.
- `Get-FileHash`: Fondamentale durante l'analisi dei malware o le indagini forensi (Incident Response). Calcola istantaneamente l'hash di un file (SHA256, MD5, ecc.) permettendo all'analista di confrontarlo immediatamente con database di threat intelligence (es. VirusTotal) per verificarne la pericolosità.
- `Get-NetTCPConnection`: L'equivalente PowerShell nativo di `netstat`. Risulta estremamente utile negli script di automazione per cercare connessioni di rete anomale (es. reverse shell) o porte in ascolto non autorizzate, permettendo di filtrare i risultati in modo dinamico come oggetti.

Questi strumenti, uniti alle capacità di scripting di PowerShell, consentono di automatizzare la raccolta di artefatti (triage) riducendo drasticamente i tempi di reazione durante un incidente informatico.

6 Conclusioni

Questo laboratorio ha confermato che PowerShell è molto più di una semplice riga di comando: è uno strumento fondamentale di scripting, automazione e diagnostica per l'ecosistema Windows. Nello specifico, si è dimostrato essenziale per:

- Monitorare il sistema: Offre una visibilità profonda sulle connessioni di rete e sui processi attivi (es. tramite l'uso di netstat e la correlazione dei PID).
- Ottimizzare le operazioni: Permette di automatizzare rapidamente compiti complessi, superando i limiti e i passaggi multipli dell'interfaccia grafica.
- Potenziare la sicurezza: Risulta indispensabile per gli analisti IT, accelerando drasticamente le attività forensi, l'ispezione dei log e le operazioni di Incident Response di fronte a potenziali minacce.